604 731

604931

1

NOTES ON THE THEORY
OF DYNAMIC PROGRAMMING—V
MAXIMIZATION OVER DISCRETE SETS

By

Richard Bellman

P-721 RH

Revised 12 December 1955

The RAND Corporation
1700 MAIN ST · SANTA MONICA · CALIFORNIA

Summary

The theory of dynamic programming is applied to a class of problems involving maximization over discrete sets. The solution is made to depend on the solution of a class of functional equations.

NOTES ON THE THEORY OF DYNAMIC PROGRAMMING—IV
MAXIMIZATION OVER DISCRETE SETS

§1. Introduction

A problem of frequent occurrence is that of determining the maximum of a function $F(x_1, x_2, \ldots, x_N)$ subject to the constraints

(1) (a) $G_i(x_1, x_2, \ldots, x_N) \leq c_i$, $i=1,2,\ldots,K$

(b) $x_i \in S_i$,

where S_i is a discrete, usually finite, set. The most important case is that where each S_i is a finite set of integers, and an interesting sub-case is that where $x_i = 0$ or 1.

A particular class of problems of this type concerns the maximization of

(2) $$F(x) = \sum_{i=1}^{N} F_i(x_i),$$

over the set of x_i constrained by the relations

(3) (a) $\sum_{j=1}^{N} G_{ij}(x_j) \leq c_i$, $i=1,2,\ldots,K$

(b) $x_i \in S_i$, $i=1,2,\ldots,K$,

with $G_{ij}(x_j) \geq 0$ for $x_j \in S_j$.

Even in the case where the F_i and G_{ij} are linear functions of the x_i, this problem at the moment escapes any of the standard computational algorithms of linear programming, such as the simplex method of G Dantzig.

We shall show that this problem may be treated by means of the functional equation technique of the theory of dynamic programming, [1], and that this technique yields a very simple computational solution whenever the number of constraints is small.

We shall also indicate the application of the method to a problem involving mutually exclusive activities. Here we have an additional constraint of the form

$$(4) \qquad x_i x_{i+1} = 0.$$

§2. Functional Equation

Let us define the sequence of functions

$$(1) \qquad f_N(c_1, c_2, \ldots, c_K) = \max_{\{x\}} \sum_{i=1}^{N} F_i(x_i),$$

where the x_i are subject to the constraints of (1.3). Then

$$(2) \qquad f_1(c_1, c_2, \ldots, c_K) = \max_{x_1} F_1(x_1)$$

where

$$(3) \qquad (a) \quad G_{11}(x_1) \leq c_1, \ldots, G_{K1}(x_1) \leq c_K,$$

$$(b) \quad x_1 \in S_1.$$

Applying the principle of optimality, we obtain the recurrence relation

(4) $$f_N(c_1, c_2, \ldots, c_K) = \underset{x_N}{\text{Max}} \left[F_N(x_N) + f_{N-1}(c_1 - G_{1N}(x_N), \ldots, c_K - G_{KN}(x_N))\right],$$

where

(5) (a) $G_{1N}(x_N) \leq c_1, \ldots, G_{KN}(x_N) \leq c_K.$

(b) $x_N \in S_N.$

§3. Example

Consider the problem of determining the maximum of $L_N(x) = \sum_{i=1}^{N} a_i x_i$ subject to the constraints

(1) (a) $\sum_{i=1}^{N} b_i x_i \leq c,$

(b) $x_i = 0$ or $1,$

where $a_i, b_i > 0.$

Here

(2) $$f_1(c) = a_1, \quad c \geq b_1,$$
$$= 0, \quad c < b_1,$$

and

(3) $$f_N(c) = \underset{x_N = 0,1}{\text{Max}} \left[a_N x_N + f_{N-1}(c - b_N x_N)\right], \quad c \geq b_N$$
$$= f_{N-1}(c), \quad c < b_N.$$

§4. Discussion

The functions $f_N(c)$ may now be computed with ease, on either a digital or hand computer, depending upon the size of the system, starting with the known value $f_1(c)$.

To give an example, suppose that $N = 50$ and $c = 200$, with the a_i, b_i integers ranging between 1 and 10. The naive approach involves the testing of 2^{50} sets of values, i.e., all possible combinations of accept or reject. Since $2^{50} \cong 10^{50(.30)} = 10^{4.5}$, this is a considerable task. Conventional linear programming techniques fail because of the restriction that the x_i be integral. For the case where $N = 50$, a roundoff of the linear programming solution may cause considerable error.

Using the above method, we must compute 50 functions $\{f_N(c),\}$ each containing 200 entries, $c = 1, 2, 3, \cdots$. If the a_i and b_i are irrational, we may have to refine the c-grid in order not to introduce round-off errors of importance. An important point to note is that doubling the size of N will double the computational time, which is to say that the time required for computing the solution in this fashion is proportional to N, rather than dependent upon N in some exponential fashion, as in ordinary search methods.

In return for the labor expended in computing the sequence $\{f_N(c),\}$ one has all the advantages of a "sensitivity analysis". It is easy to trace the influence of c and N upon the maximum value and the behavior of the maximizing $x_N = x_N(c)$.

Let us now discuss in more detail the remark we made in the introduction stating that this technique is, at the present time, restricted to problems involving a small number of constraints.

Consider a cargo-handling problem in which we have a number of items possessing values v_i, weights w_i and sizes s_i. We wish to maximize the value of the cargo carried, subject to weight restriction w and a volume restriction s.

The mathematical problem is that of maximizing

$$(1) \qquad L(x) = \sum_{i=1}^{N} x_i v_i ,$$

subject to the restrictions

$$(2) \qquad (a) \quad \sum_{i=1}^{N} x_i w_i \leq w,$$

$$(b) \quad \sum_{i=1}^{N} x_i s_i \leq s$$

$$(c) \quad x_i = 0, 1, 2, \cdots$$

Defining

$$(3) \qquad f_N(w, s) = \text{Max } L(x),$$

we readily obtain

$$(4) \qquad f_1(w, s) = v_1 \text{ Min } \left(\left[\frac{w}{w_1} \right], \left[\frac{s}{s_1} \right] \right),$$

$$f_k(w, s) = \underset{R}{\text{Max}} \left[v_k x_k + f_{k-1}(w - x_k w_k, s - x_k s_k) \right],$$

where R is the set

$$(5) \qquad x_k = 0, 1, 2, \cdots, \text{Min} \left(\left[\frac{w}{w_k} \right], \left[\frac{s}{s_k} \right] \right).$$

Taking the parameters w_i, s_i and v_i to be integers, we will, in general, be required to N functions of two variables, tabulated at the points of a grid $w = 0, 1, 2, \cdots, W$, $s = 0, 1, 2, \cdots, S$. If W and S are of the order of magnitude of 100, this requires 10^4 values. This is still within the capability of modern machines.

It is clear, however, that one additional constraint of the same type puts us in the 10^6 range. This exceeds the capability of any present day machine.

If, on the other hand, there are a large number of constraints, each with a small range, then the method is useful.

§5. Example—Mutually Exclusive Activities

Consider the problem of the preceding section under the additional constraint

$$(1) \qquad x_i x_{i+1} = 0, \qquad i = 1, 2, \cdots, N - 1$$

Define the sequence of functions

$$(2) \qquad f_N(c, b) = \underset{\{x\}}{\text{Max}} \sum_{i=1}^{N} a_i x_i$$

where the x_i are subject to the constraints

$$(3) \qquad \text{(a)} \quad x_n \cdot b = 0, \qquad b = 0 \text{ or } 1$$

$$\text{(b)} \quad \sum_{i=1}^{N} b_i x_i \le c.$$

Then we have the recurrence relation

(4) $$f_N(c, b) = \max_{x_N=0,1} \left[a_N x_N + f_{N-1}(c - b_N x_N;\ x_N)\right]$$

To determine the solution we must compute the double sequence

$\left\{f_N(c, 0),\ f_N(c, 1)\right\}$.

References

1. Bellman, R., "The Theory of Dynamic Programming", Bull. Amer. Math. Soc., Vol. 60 (1954), pp. 503-516.